What is claimed is:

1. A system for evaluating a structured message store for message redundancy, comprising:

a parser extracting a header and a message body from each of a plurality of messages maintained in a structured message store;

a digester calculating a substantially unique hash code over at least part of the header and over the message body of each message; and

a message deduper grouping the messages by the hash codes and identifying one such message as a unique message within each group.

2. A system according to Claim 1, further comprising:

a comparer grouping the messages by conversation thread, comparing the message body for each message within each conversation thread group, and identifying at least one such message within each conversation thread group as a unique message.

3. A system according to Claim 2, further comprising:

a sorter sorting the messages in each conversation thread group by length, and identifying one such message having a longest length as the unique message.

4. A system according to Claim 1, wherein a plurality of the messages each further comprise at least one attachment, for each such message, further comprising:

an attachment digester calculating a substantially unique hash code over at least part of the at least one attachment for each such message;

a comparer, comprising:

a concatenator logically concatenating the hash code for the message and the hash code for the at least one attachment; and

an attachment deduper grouping the messages by the logically concatenated hash codes, comparing the logically concatenated hash codes for each message within each group, and identifying each message with a matching logically concatenated hash code as an exact duplicate message.

5. A system according to Claim 4, further comprising:

the comparer identifying each message with one or more hash codes for the at least one attachment comprising a subset of the hash codes for the at least one attachment for at least one other message within each group as a near duplicate message.

6. A system according to Claim 5, further comprising:

the comparer identifying each message without such a subset of the hash codes as a unique message.

7. A system according to Claim 1, further comprising:

a structured database storing the messages with each message maintained in a separate indexed record in accordance with a database schema.

8. A system according to Claim 1, further comprising:

a shadow store storing the non-unique messages.

9. A system according to Claim 1, further comprising:

a log tracking message processing.

10. A system according to Claim 1, wherein each message header further comprises routing data comprising one or more data fields selected from the group comprising recipient, sender, copy-to, blind-copy-to, and subject.

11. A system according to Claim 1, wherein each hash code is calculated using a one-way function and comprises alphanumeric, numeric, and alphabetic character strings.

12. A system according to Claim 11, wherein the one-way function is selected from the group comprising at least one of the MD5 and secure hashing algorithms.

13. A method for evaluating a structured message store for message redundancy, comprising:

extracting a header and a message body from each of a plurality of messages maintained in a structured message store;

calculating a substantially unique hash code over at least part of the header and over the message body of each message; and

grouping the messages by the hash codes and identifying one such message as a unique message within each group.

14. A method according to Claim 13, further comprising:

grouping the messages by conversation thread;

comparing the message body for each message within each conversation thread group; and

identifying at least one such message within each conversation thread group as a unique message.

15. A method according to Claim 14, further comprising:

sorting the messages in each conversation thread group by length, and identifying one such message having a longest length as the unique message.

16. A method according to Claim 13, wherein a plurality of the messages each further comprise at least one attachment, for each such message, further comprising:

calculating a substantially unique hash code over at least part of the at least one attachment for each such message;

logically concatenating the hash code for the message and the hash code for the at least one attachment;

grouping the messages by the logically concatenated hash codes;

comparing the logically concatenated hash codes for each message within each group; and

identifying each message with a matching logically concatenated hash code as an exact duplicate message.

17. A method according to Claim 16, further comprising:

identifying each message with one or more hash codes for the at least one attachment comprising a subset of the hash codes for the at least one attachment for at least one other message within each group as a near duplicate message.

18. A method according to Claim 17, further comprising:

identifying each message without such a subset of the hash codes as a unique message.

19. A method according to Claim 13, further comprising:

maintaining a structured database storing the messages with each message maintained in a separate indexed record in accordance with a database schema.

20. A method according to Claim 13, further comprising:

maintaining a shadow store storing the non-unique messages.

21. A method according to Claim 13, further comprising:

maintaining a log tracking message processing.

22. A method according to Claim 13, wherein each message header further comprises routing data comprising one or more data fields selected from the group comprising recipient, sender, copy-to, blind-copy-to, and subject.

23. A method according to Claim 13, wherein each hash code is calculated using a one-way function and comprises alphanumeric, numeric, and alphabetic character strings.

24. A method according to Claim 23, wherein the one-way function is selected from the group comprising at least one of the MD5 and secure hashing algorithms.

25. A computer-readable storage medium holding code for performing the method of Claim 13.

26. An apparatus for evaluating a structured message store for message redundancy, comprising:

means for extracting a header and a message body from each of a plurality of messages maintained in a structured message store;

means for calculating a substantially unique hash code over at least part of the header and over the message body of each message; and

means for grouping the messages by the hash codes and identifying one such message as a unique message within each group.

27. A system for culling duplicative messages maintained in a structured message store, comprising:

a message extractor retrieving a plurality of messages maintained in a structured message store with each message comprising a header and a message body;

a digester calculating a substantially unique hash code over at least part of the header and over the message body;

a message deduper, comprising:

a comparer comparing the hash codes for each message within each group; and

a culling module culling each message having an hash code matching the hash code for at least one other message within the group and retaining one such non-culled message as a unique message.

28. A system according to Claim 27, wherein each such non-culled message is retained as a potential unique message, further comprising:

a comparer grouping the potential unique messages by conversation thread and comparing the message body for each potential unique message within each conversation thread group; and

a culling module culling each potential unique message having a message body contained within at least one other message within each group and retaining one such non-culled message as a unique message.

29. A system according to Claim 27, further comprising:

a sorter sorting the potential unique messages within each group by conversation thread.

30. A system according to Claim 27, wherein a plurality of the messages each further comprise at least one attachment, further comprising:

the digester calculating a substantially unique hash code over at least part of the at least one attachment for each message, combining the hash code for each message and the hash code for the at least one attachment, and grouping the messages by the combined hash codes;

the comparer comparing the combined hash codes for each message within each group; and

the culling module culling each message with a matching combined hash codes and retaining one such non-culled message as a unique message.

31. A system according to Claim 30, further comprising:

the comparer identifying each message with one or more hash codes for the at least one attachment comprising a subset of the hash codes for the at least one attachment for at least one other message within each group; and

the culling module culling each message with such a subset of the hash codes and retaining one such non-culled message as a unique message.

32. A method for culling duplicative messages maintained in a structured message store, comprising:

retrieving a plurality of messages maintained in a structured message store with each message comprising a header and a message body;

calculating a substantially unique hash code over at least part of the header and over the message body;

comparing the hash codes for each message within each group; and

culling each message having an hash code matching the hash code for at least one other message within the group; and

retaining one such non-culled message as a unique message.

33. A method according to Claim 32, wherein each such non-culled message is retained as a potential unique message, further comprising:

grouping the potential unique messages by conversation thread;

comparing the message body for each potential unique message within each conversation thread group; and

culling each potential unique message having a message body contained within at least one other message within each group and retaining one such non-culled message as a unique message.

34. A method according to Claim 32, further comprising:

sorting the potential unique messages within each group by conversation thread.

35. A method according to Claim 32, wherein a plurality of the messages each further comprise at least one attachment, further comprising:

calculating a substantially unique hash code over at least part of the at least one attachment for each message;

combining the hash code for each message and the hash code for the at least one attachment;

grouping the messages by the combined hash codes;

comparing the combined hash codes for each message within each group;

culling each message with a matching combined hash codes; and

retaining one such non-culled message as a unique message.

36. A method according to Claim 35, further comprising:

identifying each message with one or more hash codes for the at least one attachment comprising a subset of the hash codes for the at least one attachment for at least one other message within each group; and

culling each message with such a subset of the hash codes and retaining one such non-culled message as a unique message.

37. A computer-readable storage medium holding code for performing the method of Claim 32.

38. An apparatus for culling duplicative messages maintained in a structured message store, comprising:

means for retrieving a plurality of messages maintained in a structured message store with each message comprising a header and a message body;

means for calculating a substantially unique hash code over at least part of the header and over the message body;

means for comparing the hash codes for each message within each group; and

means for culling each message having an hash code matching the hash code for at least one other message within the group; and

means for retaining one such non-culled message as a unique message.